1 First Order Logic

1. ~~Vibe~~ Vocab check: are you familiar with the following terms?

   (a) Objects

   Things that can be described with relations, and can be returned from functions

   (b) Relations

   Sentences that describe relationships between objects, e.g. Fatherof(a, b). Returns a boolean.

   (c) Functions

   Describe an object using another object, e.g. Fatherof(a). Returns an object.

   (d) Constants

   Specific objects in the world

   (e) Variables

   Lower case letters that can take the value of different objects

   (f) Interpretation

   A mapping of variables to objects

   (g) Connectives

   Describe relationships between sentences (same as operators in propositional logic: $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$)

   (h) Equality

   Given an interpretation, two terms are equal if they are mapped to the same object

   (i) Quantifiers

   $\exists, \forall$

   (j) Atomic Sentence

   Sentences consisting of only one relation (no connectives)

   (k) Unification

   Binding of variables with known objects with some substitution θ (similar to interpretation)

2. Which of the following FOL sentences correctly expresses its corresponding English sentence?

   (a) There was a student at CMU who never did 281 homework but passed the class.
   $\exists x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \implies \text{Pass}(x, 281)$

   False. This sentence is true as long as there exists an x who is not a student at CMU or does 281 homework. This should instead be $\exists x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \wedge \text{Pass}(x, 281)$.

   (b) If a student likes Pat, they'll pass the class.
   $\forall x, \text{Student}(x) \wedge \text{Likes}(x, \text{Pat}) \implies \text{Pass}(x, 281)$

   True.

   (c) All students at CMU who never did 281 homework passed the class.
   $\forall x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \wedge \text{Pass}(x, 281)$

   False - this should instead be $\forall x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \implies \text{Pass}(x, 281)$

3. Which of the following is true with respect to the English sentences (a) and (c) above?

   (a) $\models$ (c) (c) $\models$ (a) Both Neither

Using the correct formulations of (a) and (c), where
(a): $\exists x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \wedge \text{Pass}(x, 281)$
(c): $\forall x, \text{IsStudent}(x, \text{CMU}) \wedge \neg\text{DoesHW}(x, 281) \implies \text{Pass}(x, 281)$
neither (a) $\models$ (c) nor (c) $\models$ (a) are true.

4. Forward Chaining

   (a) What are the requirements for Knowledge Base in Forward Chaining?

   KB can only contain definite clauses (i.e., clauses with exactly one positive literal).

   (b) What does FOL Forward Chaining return?

   Given a knowledge base KB and sentence α, FOL-FC returns a unification θ for α, or false if no unification is possible.

2 Planning

1. Vocab check: are you familiar with the following terms?

   (a) Predicates

   Conditions that the current state of the world must satisfy to perform an operation.

   (b) Operators

   Perform actions that change the state of the world

   (c) Linear Planning

   Solve one goal at a time.

   (d) Non-linear Planning

   Interleave goals to achieve plans

   (e) Inconsistency

   The effects of two actions negate each other

   (f) Interference

   One action deletes/negates a precondition of the other

   (g) Competing Needs

   The preconditions of two actions negate each other

   (h) Complete

   Can always find a solution whenever one exists

   (i) Sound

   All solutions found are legal.

   (j) Optimal

   Shortest path to goal

2. What are in the knowledge base of logic agents and classical planning problem, respectively?

   Logical agents: Symbols and Implications
   Planning: Predicates and operators

3. What are the 3 components when defining an operator?

   Preconditions, add list, delete list (you could technically define in two: Preconditions, add list)

4. Is linear planning complete? Optimal? What about non-linear planning?

   Linear Planning: Not complete, one goal can immediately undo the other, and the plan stucks there. Not optimal: The plan naively attempts in order where there can be shortcuts among some goals. Non-linear Planning: Complete and optimal. It performs search and would always reach the depth of the solution whenever there exists one. It returns the solution as soon as one is found so there can't be any shorter path (if all interleavings are searched).

5. The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at A, the bananas at B, and the box at C. The monkey and box have height *Low*, but if the monkey climbs onto the box he will have height *High*, the same as the bananas. The actions available to the monkey include *Go* from one place to another, *Push* an object from one place to another, *ClimbUp* onto or *ClimbDown* from an object, and *Grasp* or *Ungrasp* an object. The result of a *Grasp* is that the monkey holds the object if the monkey and object are in the same place at the same height. We want to formulate this problem using GraphPlan.

   (a) Define the initial and goal states.

Initial state:
At(Monkey,A), At(Bananas,B), At(Box,C), Height(Monkey,Low), Height(Box,Low), Height(Bananas,High), Pushable(Box), Climbable(Box)

Goal state:
Have(Monkey, Bananas)

(b) Write the six action schemas, i.e., define the preconditions and effects for each possible action the monkey can take.

i. *Action*(Go(x,y))

   A. **Preconditions:** At(Monkey,x)
   B. **Add List:** At(Monkey,y)
   C. **Delete List:** At(Monkey,x)

ii. *Action*(Push(b,x,y))

   A. **Preconditions:** At(Monkey,x)
   B. **Add List:** At(b,y), At(Monkey,y)
   C. **Delete List:** At(b,x), At(Monkey,x)

iii. *Action*(ClimbUp(b))

   A. **Preconditions:** At(Monkey,x), At(b,x), Climbable(b)
   B. **Add List:** On(Monkey,b)
   C. **Delete List:** Height(Monkey,High)

iv. *Action*(Grasp(b))

   A. **Preconditions:** Height(Monkey,h), Height(b,h), At(Monkey,x), At(b,x)
   B. **Add List:** Have(Monkey,b))

v. *Action*(ClimbDown(b))

   A. **Preconditions:** On(Monkey,b), Height(Monkey,High)
   B. **Add List:** Height(Monkey,Low)
   C. **Delete List:** On(Monkey,b), Height(Monkey,High)

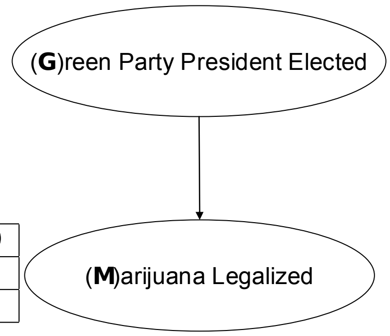
vi. *Action*(UnGrasp(b))

   A. **Preconditions:** Have(Monkey,b)
   B. **Delete List:** Have(Monkey,b))

3 Bayes' Nets: Green Party President

It's election year again! In a parallel universe the Green Party is running for presidency. Pundits believe that Green Party presidents are more likely to legalize marijuana than candidates from other parties, but legalization could occur under any administration. Armed with the power of probability, the analysts model the situation with the Bayes Net below.

	+g	-g
P(G)	0.1	0.9



	P(+m | G)	P(-m | G)
+g	0.667	0.333
-g	0.25	0.75

1. Fill in the joint probability table over G and M.

G	M	P(G, M)
+g	+m	
+g	-m	
-g	+m	
-g	-m	

G	M	P(G, M)
+g	+m	1/15
+g	-m	1/30
-g	+m	9/40
-g	-m	27/40

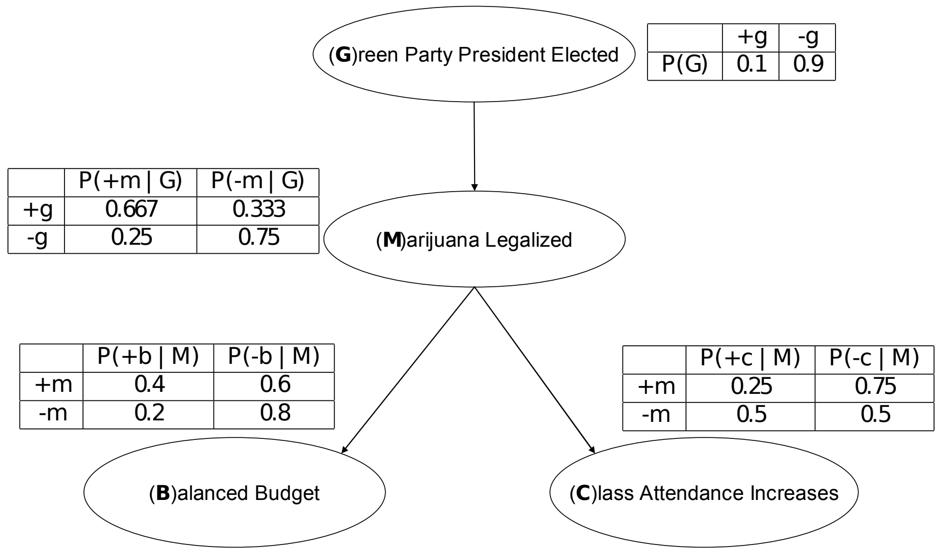
2. What is P(+m), the marginal probability that marijuana is legalized?

$$\mathrm{P}(+\mathrm{m}) = \mathrm{P}(+\mathrm{m}, +\mathrm{g}) + \mathrm{P}(+\mathrm{m}, -\mathrm{g}) = \mathrm{P}(+\mathrm{m} \mid +\mathrm{g})\mathrm{P}(+\mathrm{g}) + \mathrm{P}(+\mathrm{m} \mid -\mathrm{g})\mathrm{P}(-\mathrm{g}) = \frac{2}{3} \cdot \frac{1}{10} + \frac{1}{4} \cdot \frac{9}{10} = \frac{7}{24}$$

3. News agencies air 24/7 coverage of the recent legalization of marijuana (+m), but you can't seem to find out who won the election. What is the conditional probability P(+g | +m) that a Green Party president was elected?

$$\mathrm{P}(+\mathrm{g} \mid +\mathrm{m}) = \frac{\mathrm{P}(+\mathrm{g}, +\mathrm{m})}{\mathrm{P}(+\mathrm{m})} = \frac{\mathrm{P}(+\mathrm{m} \mid +\mathrm{g})\mathrm{P}(+\mathrm{g})}{\mathrm{P}(+\mathrm{m})} = \frac{\frac{2}{3} \cdot \frac{1}{10}}{\frac{7}{24}} = \frac{8}{35}$$

We can make better inferences if we observe more evidence. On the next page, we will expand on the model (Bayes net) by introducing two new random variables: whether the budget is balanced (B), and whether class attendance increases (C). The expanded Bayes net and conditional distributions are shown below.



	+g	-g
P(G)	0.1	0.9

	P(+m | G)	P(-m | G)
+g	0.667	0.333
-g	0.25	0.75

	P(+b | M)	P(-b | M)
+m	0.4	0.6
-m	0.2	0.8

	P(+c | M)	P(-c | M)
+m	0.25	0.75
-m	0.5	0.5

4. The full joint distribution is given below. Fill in the missing values.

G	M	B	C	$P(G,M,B,C)$	G	M	B	C	$P(G,M,B,C)$
+	+	+	+	1/150	-	+	+	+	9/400
+	+	+	-		-	+	+	-	27/400
+	+	-	+	1/100	-	+	-	+	27/800
+	+	-	-	3/100	-	+	-	-	81/800
+	-	+	+	1/300	-	-	+	+	27/400
+	-	+	-	1/300	-	-	+	-	27/400
+	-	-	+		-	-	-	+	
+	-	-	-	1/75	-	-	-	-	27/100

G	M	B	C	$P(G,M,B,C)$	G	M	B	C	$P(G,M,B,C)$
+	+	+	+	1/150	-	+	+	+	9/400
+	+	+	-	1/50	-	+	+	-	27/400
+	+	-	+	1/100	-	+	-	+	27/800
+	+	-	-	3/100	-	+	-	-	81/800
+	-	+	+	1/300	-	-	+	+	27/400
+	-	+	-	1/300	-	-	+	-	27/400
+	-	-	+	1/75	-	-	-	+	27/100
+	-	-	-	1/75	-	-	-	-	27/100

5. Compute the following quantities. You may use either the full joint distribution or the conditional tables, whichever is more convenient.

   (a) P(+b | +m) =

   $\frac{4}{10}$ (directly from conditional)

   (b) P(+b | +m, +g) =

   $\frac{4}{10}$ (also directly from conditional, since $B \perp\!\!\!\perp G | M$)

   (c) P(+b) =

$\sum\limits_{G,M,C} \text{P(G,M,+b,C)} = \frac{8}{1200} + \frac{24}{1200} + \frac{4}{1200} + \frac{4}{1200} + \frac{27}{1200} + \frac{81}{1200} + \frac{81}{1200} + \frac{81}{1200} = \frac{31}{120}$

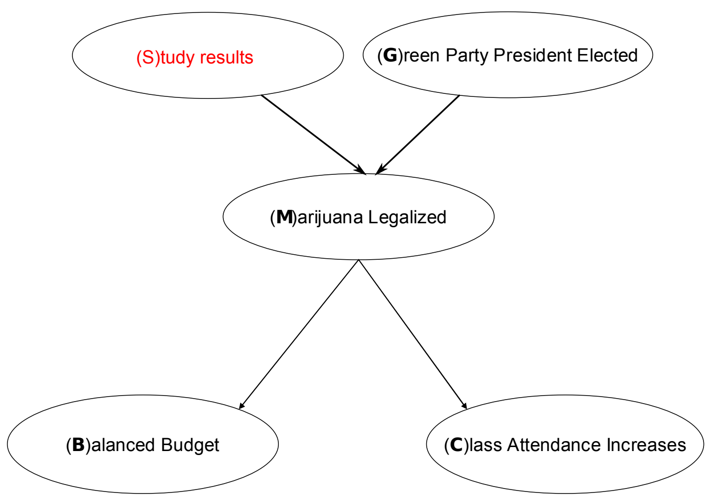
(summed from joint)

(d) P(+c | +b) =

$\frac{\text{P(+b, +c)}}{\text{P(+b)}} = \frac{\sum\limits_{G,M} \text{P(G,M,+b,+c)}}{31/120} = \left(\frac{8}{1200} + \frac{4}{1200} + \frac{27}{1200} + \frac{81}{1200}\right) \cdot \frac{120}{31} = \frac{12}{31}$

(summed from joint)

6. Now, add a node S to the Bayes net above that reflects the possibility that a new scientific study could influence the probability that marijuana is legalized. Assume that the study does not directly influence B or C. Which CPT(s) need to be modified?



$P(M|G)$ will become $P(M|G,S)$, and will contain 8 entries instead of 4.

7. Consider your augmented model. Just based on the structure, which of the following are guaranteed to be true?

(a) $B \perp\!\!\!\perp G$

(b) $C \perp\!\!\!\perp G \mid M$

(c) $G \perp\!\!\!\perp S$

(d) $G \perp\!\!\!\perp S \mid M$

(e) $G \perp\!\!\!\perp S \mid B$

(f) $B \perp\!\!\!\perp C$

(g) $B \perp\!\!\!\perp C \mid G$

(b), (c)

4 LP/IP/Optimization

1. Warm Up Questions

   (a) What is the relationship between branch and bound and pruning?

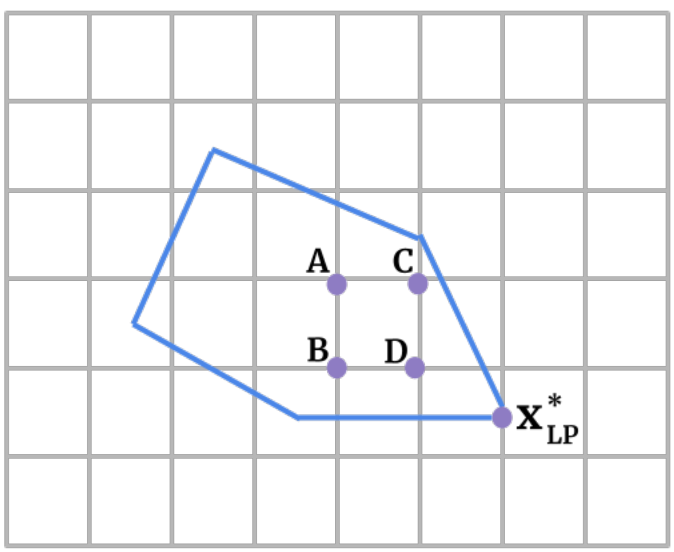
   In branch and bound, we stop recurring down a branch when the LP returns a worse objective than the best feasible IP objective seen so far. This is similar to pruning in Minimax because we know we can do better by taking another branch, so we prune.

   (b) T or F: As the magnitude of c increases, the distance between the contour lines of the objective $c^T x$ increases as well. Why or why not?

   False - an increase in the magnitude of c implies that a lesser distance needs to be traversed in order to incur the same increase in cost. This means that the distance between contour lines actually decreases.

2. Consider the integer programming problem illustrated by the figure below.

   - The blue lines represent the boundaries of the feasible region and the gray vertical and horizontal lines represent the integer values for each axis.
   - Let x^*_{LP} be the unique point that minimizes the relaxed linear program. It happens to lie on a vertical gray line and a horizontal blue line.
   - Let x^*_{IP} (unlabeled) be the unique point that minimizes the integer program.
   - When running branch and bound, we will explore the x_i less than constraint subtree before the x_i greater than constraint subtree ("less" being to the left and down).



   (a) Which of the points (A, B, C, or D) can possibly be x^*_{IP}?

   C or D
   Since x^*_{LP} is the unique point that minimizes the related linear program and not the vertex directly left of it, we know that the horizontal component of the cost vector will point left. Thus, C will be of lower cost than A, and D will be of lower cost than B. This means that A or B can never be the optimal integer solution.

   (b) Which of the points from your answer to part (a) needs to be x^*_{IP} in order for branch and bound to find the IP solution at the minimum possible depth?

D

If we know that D is the unique point that minimizes the integer program, we know that the vertical component of the cost vector has to point up. If we run branch and bound knowing that, we find the optimal IP solution at depth 2.

However, if we know that C is the unique point that minimizes the integer program, we know that the vertical component of the cost vector has to point down. If we run branch and bound, we find the optimal IP solution at a depth greater than 2.

5 Reinforcement Learning

1. Consider ϵ-greedy search on a deterministic environment. We want ϵ to be small so we act according to our learned policy. Suppose we start with $\epsilon = 0$ and don't change it. Are we guaranteed to learn the optimal policy? Why or why not?

   We are not guaranteed to learn the optimal policy because we will never explore potentially better actions and states.

2. Now suppose we set $\epsilon = 1$ and don't change it. Are we guaranteed to learn the optimal policy? Why/why not?

   We will learn the optimal policy because we will eventually explore all actions and states. However, we will never act according to the optimal policy because we always move randomly. This method will accumulate more regret than $\epsilon < 1$.
   This is the idea behind ϵ-greedy search with decaying ϵ: explore at the start to learn, then slowly begin acting according to the optimal policy.

6 MDPs

1. Concepts:

   (a) What does the Markov Property state?

   Markov Property states that action outcomes depend on the current state only. It states that action outcomes do not depend on the past.

   (b) What are the Bellman Equations, and when are they used?

   The Bellman Equations give a definition of "optimal utility" via expectimax recurrence. They give a simple one-step lookahead relationship amongst optimal utility values.

   (c) What is a policy? What is an optimal policy?

   A policy is a function that maps states to actions. $\pi(s)$ gives an action for state s. An optimal policy is a policy that maximizes the expected utility if an agent follows it.

   (d) How does the discount factor γ affect the agent's policy search? Why is it important?

   γ determines how much the value of a state should take into account future states. The higher the discount factor, the more one state would value distant states. Having $0 < \gamma < 1$ also helps our algorithms converge.

   (e) What are the two steps to Policy Iteration?

   Policy evaluation and policy improvement.

   (f) What is the relationship between $V^*(s)$ and $Q(s, a)$?

   $V^*(s) = \max_a Q(s, a)$

2. In a certain country there are N cities, all connected by roads in a circular fashion. A wandering poet is travelling around the country and staging shows in its different cities. She can choose to move from a city to each of the neighboring ones or she can stay in her current city i and perform, getting a reward r_i. If she chooses to travel, she will have a success probability of p_i. There is a $1 - p_i$ chance she will encounter a dragon along the way, which means she will have to turn back and wait the next day. If she is successful in travelling, she gains a reward of 0 for the day. And if she is unsuccessful at travelling, she can still perform a little bit when she gets back, giving her a reward of $r_i/2$.

   Let $r_i = 1$ and $p_i = 0.5$ for all i and let $\gamma = 0.5$. For $1 \leq i \leq N$, answer the following questions with concrete numbers:

   i) What is the value $V^{\text{stay}}(i)$ under the policy the wandering poet always chooses to stay?

   $$V^{\text{stay}}(i) = r_i + \gamma V^{\text{stay}}(i) = 1 + 0.5V^{\text{stay}}(i)$$
   $$V^{\text{stay}}(i) = 2$$

   ii) What is the value $V^{\text{next}}(i)$ under the policy the wandering poet always chooses to go to the next city? You may assume that the values of each state converge to the same value.

   $$V^{\text{next}}(i) = 0.5(\frac{1}{2} + 0.5V^{\text{next}}(i)) + 0.5(0 + 0.5V^{\text{next}}(i + 1))$$
   $$V^{\text{next}}(i) = V^{\text{next}}(i + 1) = \frac{1}{2}$$